

# Lexical Features are Not Necessary for Sequence Labeling

## Abstract

We use the technique of SVM anchoring to demonstrate that lexical features extracted from an annotated training corpus are not necessary to obtain state of the art results on tasks such as Named Entity Recognition and Chunking. While standard models require as many as 100K distinct features, we derive models with as little as 1K features that perform as well. These robust reduced models indicate that the way rare lexical features contribute to classification in NLP is not yet well understood. As a general strategy, we believe lexical features should not be directly derived from a training corpus but instead, carefully inferred and selected from other sources.

## 1 Introduction

Common NLP tasks, such as Named Entity Recognition and Chunking, involve the identification of spans of words belonging to the same phrase. These tasks are traditionally reduced to a tagging task, in which each word is to be classified as either **B**eginning a span, **I**nside a span, or **O**utside of a span. The decision is based on the word to be classified and its neighbors. Features supporting the classification usually include the word forms themselves and properties derived from the word forms, such as prefixes, suffixes, capitalization information, and parts-of-speech. It is a widely accepted that the lexical information (word forms) is crucial for building accurate systems for these tasks. Indeed, all the better-performing systems in the CoNLL shared tasks competitions for Chunking (Sang and Buchholz, 2000) and Named Entity Recognition (Tjong

Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) make extensive use of such lexical information.

Is this belief justified? In this paper, we show that the influence of lexical features on such sequence labeling tasks is more complex than is generally assumed. We find that exact word forms aren't necessary for accurate classification. This is important because relying on the exact word forms that appear in a training corpus leads to over-fitting.

In this work, we focus on learning with Support Vector Machines (SVMs) (Vapnik, 1995). SVM classifiers can handle very large feature spaces, and produce state-of-the-art results for NLP applications (see e.g. (Kudo and Matsumoto, 2000; Nivre et al., 2006)). Alas, when trained on pruned feature sets, in which rare lexical items are removed, SVM models suffer a loss in classification accuracy. It would seem that rare lexical items are indeed crucial for SVM classification performance. However, Goldberg and Elhadad (2007) suggest that the SVM learner is using the rare lexical features for singling out hard cases rather than for learning meaningful generalizations. We provide further evidence to support this claim in this paper.

We show that by using a variant of SVM – *Anchored SVM Learning* (Goldberg and Elhadad, 2007) with a polynomial kernel, we can learn accurate models for English NP-chunking (Marcus and Ramshaw, 1995), base-phrase chunking (CoNLL 2000), and Dutch Named Entity Recognition (CoNLL 2002), on a heavily pruned feature space. Our models make use of only a fraction of the lexical features available in the training set (less than 1%), and yet provide

highly-competitive accuracies.

This goes to show that with an appropriate learning method, orthographic and structural (in the form of POS tag sequences) information is mostly sufficient for achieving state-of-the-art performance on these kind of sequence labeling tasks. We believe this data motivates a different strategy to incorporate lexical features into classification models: instead of collecting the raw lexical forms appearing in a training corpus, we should attempt to actively construct a feature space including lexical features derived from external sources.

## 2 Learning with Less Features

We adopt the common feature representation in which each data-point is represented as a sparse  $D$  dimensional binary-valued vector  $f$ . Each of the  $D$  possible features  $f_i$  is an indicator function. The indicator functions look at properties of the current or neighbouring words. An example of such function  $f_i$  is 1 iff the previous word-form is DOG, 0 otherwise. The lexical (word-form) features result in extremely high-dimensional (yet very sparse) feature vectors – each word-form in the vocabulary of the training set correspond to (at-least) one indicator function.

Due to the Zipfian distribution of language data, many of the lexical features are very rare, and appear only a couple times in the training set. Ideally, we would like our classifiers to learn only from robust features: consider only features that appear at least  $k$  times in the training data (rare-feature pruning). These features are more likely to appear in unseen test data, and thus such features can support more robust generalization.

However, we find empirically that performing such feature pruning *prior* to learning SVM models hurts the performance of the learned models. Our intuition is that this sensitivity to rare lexical features is not explained by the richness of information such rare features bring to the model. Instead, we believe that rare lexical features help the classifier because they make the data *artificially* more separable. To demonstrate this claim, we experiment with anchored SVM, which introduces artificial mechanical anchors into the model to achieve separability, and make rare lexical features unnecessary.

## 3 Learning Method

SVM are discriminative, max-margin, linear classifiers (Vapnik, 1995), which can be kernelized. For the formulation of SVMs in the context of NLP applications, see (Kudo and Matsumoto, 2001). SVMs with a polynomial kernel of degree 2 were shown to provide state-of-the-art performance in many NLP application, see for example (Kudo and Matsumoto, 2000; Nivre et al., 2006; Isozaki and Kazawa, 2002).

SVMs cope with inseparable data by introducing a *soft-margin* – allowing some of the training instances to be classified incorrectly subject to a penalty, controlled by a parameter  $C$ .

**Anchored SVM.** As we show in Section 4, the soft-margin heuristic performs sub-optimally for NLP tasks when the data is inseparable. We use instead the *Anchored Learning* heuristic, introduced in (Goldberg and Elhadad, 2007). The idea behind anchored learning is that some training instances are inherently ambiguous. This ambiguity stems from ambiguity in language structure, which cannot be resolved with a given feature representation. When a data-point cannot be classified, it might be due to missing information, which is not available in the data representation. Instead of allowing ambiguous items to be misclassified during training, we make the training data artificially separable. This is achieved by adding a unique feature to each training example (an *anchor*). These anchor features cause each data-point to be slightly more similar to itself than to any other data point. At test time, we remove anchor features.

In terms of kernel-based learning, anchored learning can be achieved by redefining the dot product between two vectors to take into account the identity of the vectors:  $x_i \cdot_{anc} x_j = x_i \cdot x_j + \delta_{ij}$ .

The classifier learned over the anchored data takes into account the fine interactions between the various inseparable data points. In our experiments, SVM models over anchored data have many more support vectors than soft-margin SVM models. However, the anchored models generalize much better when less features are available.

## 4 Experiments

How important are the rare lexical features for learning accurate NLP models? To investigate this question, we experiment with 3 different NLP sequence-

labeling tasks. For each task, we train a sequence of polynomial kernel ( $d=2$ ) SVM classifiers, using both soft-margin ( $C=1$ ) and anchored SVM. Each classifier is trained on a pruned feature set, in which only features appearing at least  $k$  times in the training data are kept. We vary the pruning parameter  $k$ . Pruning is performed over all the features in the model, but lexical features are most affected by it.

For all the models, we use the B-I-O representation, and perform multiclass classification using pairwise-voting. For our features, we consider properties of tokens in a 5-token window centered around the token to be classified, as well as the two previous classifier predictions. Results are reported as F-measure over labeled identified spans.

#### 4.1 Named Entity Recognition (NER)

We use the Dutch data set from the CoNLL 2002 shared task (Tjong Kim Sang, 2002). The aim is to identify named entities (persons, locations, organizations and miscellaneous) in text. The task has two stages: identification of the entities, and classification of the identified entities into their corresponding types. We focus here on the identification task.

**Features:** We use the following properties for each of the relevant tokens: word-form, POS, ORT, prefix1, prefix2, prefix3, suffix1, suffix2, suffix3. The ORT feature can take one of the following values: {number, contains-digit, contains-hyphen, capitalized, all-capitalized, URL, punctuation, regular}.

PRUNING	#FEATURES	SOFT-MARGIN	ANCHORED
0	186,421	90.92	90.78
100	5,804	90.73	90.75
1000	1,207	88.56	90.10
1500	821	85.92	89.29

Table 1: Named Entity Identification results (F-score) on dev set, with various pruning thresholds.

**Results** are presented in Table 1. Without feature pruning, we achieve an F-score of 90.9. This dataset proved to be quite resilient to feature pruning. Pruning features appearing less than 100 times results in just a slight decrease in F-score. Extremely aggressive pruning, keeping only features appearing more than 1,000 or 1,500 times in the training data, results in a big drop in F-score for the soft-margin SVM (from about 91 to 86). Much less so for the Anchored-SVM. Using Anchored SVM we achieve an F-score of 90.1 after pruning with  $k = 1,000$ .

This model has 1207 active features, and 27 unique active lexical forms.

#### 4.2 NP Chunking

The goal of this task (Marcus and Ramshaw, 1995) is the identification of non-recursive NPs. We use the data from the CoNLL 2000 shared task: NP chunks are extracted from Sections 15-18 (train) and 20 (test) of the Penn WSJ corpus. POS tagged are automatically assigned by the Brill Tagger.

**Features:** We consider the POS and word-form of each token.

PRUNING	#FEATURES	SOFT-MARGIN	ANCHORED
0	92,805	94.12	94.08
1	46,527	93.78	94.09
2	32,583	93.58	94.00
5	18,092	93.42	94.01
10	10,812	93.00	93.98
20	5,952	92.48	93.92
50	2,436	92.33	93.96
100	1,168	91.94	93.83

Table 2: NP-Chunking results (F-score), with various pruning thresholds.

**Results** are presented in Table 2. Without feature pruning ( $k = 0$ ), the soft-margin SVM performs slightly better than the Anchored-SVM. Either of the results are state-of-the-art for this task. However, even modest pruning ( $k = 2$ ) hurts the soft-margin model significantly. Not so for the anchored-SVM. Even with relatively aggressive pruning ( $k = 100$ ), the anchored model still achieves an impressive F-score of 93.83. Remarkably, in that last model, there are only 1,168 active features, and only 209 unique active lexical forms.

#### 4.3 Chunking

The goal of the Chunking task (Sang and Buchholz, 2000) is the identification of an assortment of linguistic base-phrases. We use the data from the CoNLL 2000 shared task.

**Features:** We perform two experiments. In the first experiment, we consider the POS and word-form of each token. In this setting, feature pruning resulted in a bigger loss in performance than in the two previous tasks. Error analysis reveals that many errors are due to tagger errors, especially of the present participle forms. This led us to the second experiment, in which we replaced the explicit word-forms by 2- and 3-letter suffixes.

PRUNING	#FEATURES	SOFT-MARGIN	ANCHORED
0	92,837	93.44	93.40
1	46,557	93.20	93.32
2	32,614	93.10	93.31
5	18,126	92.89	93.29
10	10,834	92.73	93.23
20	5,975	92.18	93.16
50	2,463	91.80	92.89
100	1,180	90.94	92.56

Table 3: Chunking results (F), with various pruning thresholds. Experiment 1. Features: POS, Word-Form.

PRUNING	#FEATURES	SOFT-MARGIN	ANCHORED
0	19,910	93.25	93.23
100	2,563	92.87	93.18
250	1,508	92.40	92.87

Table 4: Chunking results (F), with various pruning thresholds. Experiment 2. Features: POS, Suff2, Suff3.

**Results** are presented in Tables 3 and 4. In the first experiment (POS + word-form), the non-pruned soft-margin model is the same system as the top-performing system in the original shared task, and yields state-of-the-art results. Unlike the NP-chunking case, here feature pruning have a relatively large impact on the results even for the anchored models. However, the anchored models are still far more robust than the soft-margin ones. With  $k = 100$  pruning, the soft-margin model suffers a drop of 2.5 F points, while the anchored model suffers a drop of only 0.84 F points. Even after this drop, the anchored  $k = 100$  model still performs above the top-third system in the CoNLL 2000 shared task. This anchored  $k = 100$  model has 1,180 active features, and only 209 unique active lexical features.

The second experiment (POS + suffix3 + suffix2) is much less lexicalized to begin with. Only the word’s POS, last-2 and last-3 letters are used as features. This gives us the complete word form of many function words, and a reasonable amount of morphological marking. Surprisingly, this information proves to be quite robust. Without feature pruning, both the anchored and soft-margin model achieve near state-of-the-art performance of 93.25F. Pruning with  $k = 100$  hurts the result of the soft-margin model, but the anchored model remains robust with an F-score of 93.18. This last model has 2,563 active features. With further pruning ( $k = 250$ ), the result of the anchored model drops to 92.87F (still 3rd

place in the CoNLL shared task), with only 1,508 active features in the model.

## 5 Discussion

For all the tasks, the anchored-SVM proved to be much more robust to feature pruning than the soft-margin SVM. The experiments support our claim that rare lexical features do not provide substantial information to the model, but instead play a role in maintaining separability. When this role is taken over by anchoring, we can obtain the same level of performance with very few robust lexical features. We do not claim that lexical information is not needed. There is a significant difference between the pruned and non-pruned models for the chunking task. The anchored models are not as compact as the soft-margin ones, and contain about twice the amount of support vectors. However, the relatively high classification accuracies achieved with the heavily pruned anchored-SVM models sheds new light on the actual role of lexical features.

## References

- Y. Goldberg and M. Elhadad. 2007. SVM Model Tampering and Anchored Learning: A Case Study in Hebrew. NP Chunking. In *ACL2007*.
- H. Isozaki and H. Kazawa. 2002. Efficient Support Vector Classifiers For Named Entity Recognition. In *COLING2002*.
- T. Kudo and Y. Matsumoto. 2000. Use of Support Vector Learning for Chunk Identification. In *CoNLL-2000*.
- Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *NAACL '01*.
- M. Marcus and L. Ramshaw. 1995. Text Chunking Using Transformation-Based Learning. In *3rd ACL Workshop on Very Large Corpora*.
- J. Nivre, J. Hall, and J. Nillson. 2006. MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *LREC2006*.
- Erik F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 shared task: chunking. In *CoNLL-2000*.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *CoNLL-2003*.
- Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In *CoNLL-2002*.
- V. Vapnik. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc.